The Hong Kong Polytechnic University
Department of Electronic and Information Engineering

EIE565 AMT Laboratory 1:

Face Image Analysis and Representation Using Principal Component Analysis (PCA)

Objectives:

In this laboratory exercise, you are given 20 face images, which are used as training samples for PCA. You are required to produce the eigenfaces of the training samples, which are then used to form an eigenspace for face representation and recognition based on 4 testing images (2 face images and 2 non-face images). Through this exercise, you will learn the following:

1. representation of a face image as a high-dimensional vector;
2. generation of the principal components of a set of training images;
3. approximation of images using the principal components; and
4. face recognition using PCA.

Software Tools: MATLAB is used throughout this laboratory. You may refer to the HELP menu for the MATLAB commands used in this laboratory.

Procedures:

(a) *Reading the Training Images*:

The 20 training face images, which are aligned with the position of the two eyes and normalized to a size of 64(W)×72(H), are stored in the folder "Training". Some of these images are shown in Fig. 1.

1.bmp 2.bmp 3.bmp 4.bmp 5.bmp

Figure 1. Some of the training images.

The filenames of these image files are "*i*.bmp", where i = 1, …, 20. Run the following MATLAB codes to read the training face images:

```
clear;
NumOfSamples = 20;
str_Path = 'c:\AMT\Training\';
for i = 1: NumOfSamples
   str_Load = strcat(str_Path, num2str(i), '.bmp');
   Image = imread(str_Load);
   TrainingImage(:,i) = double(reshape(Image, [ ], 1));
end
```

What is the dimension of TrainingImage? What do the columns of this matrix represent? You may use the command "size(A)" to check the size of A.

(b) *Calculating the Mean Face and the Demeaned Faces*:

You can calculate the mean of the training face images using the following MATLAB codes:

```
for i = 1: NumOfSamples
   MeanFace = MeanFace+TrainingImage(:,i);
end
MeanFace = MeanFace/NumOfSamples;
```

The demeaned faces of the training samples can be computed as follows:

```
for i = 1: NumOfSamples
   DemeanFace(:,i) = TrainingImage(:,i) – MeanFace;
end
```

Display the mean face image and the demeaned face images. Note that the demeaned images can be displayed with grayscale bars using the following MATLAB codes:

```
for i = 1: NumOfSamples
  Display = DemeanFace(:,i);
  Display = reshape(Display, [ImageHeight ImageWidth]);
  figure(i), imagesc(Display), colorbar, colormap(gray), title('Demeaned Face');
end
```

You may save the figure in a particular image file format. Check with the tutors if you don't know how to do it.

(c) *Computing the Eigenvalues and Eigenvectors/Eigenfaces*:

(i) With the demeaned face images DemeanFace, write a MATLAB code to compute the covariance matrix. Denote this covariance matrix as CovFace1. Note that the transpose of a vector x is x' in MATLAB.

What is the dimension of CovFace1?

(ii) The MATLAB command [EV, ED] = eig(A) computes the eigenvectors and eigenvalues of the square matrix A and stores them in EV and ED, respectively. Use this command to compute the eigenvalues and eigenvectors of CovFace1. What observation can you see? Discuss your observation.

(iii) Use another method which computes the eigenvalues and eigenvectors with a smaller covariance matrix.

How many non-zero eigenvectors or non-zero eigenvalues are available? Discuss your answer.

Display the eigenvectors in image form according to the eigenvalues in descending order.

(d) *Image Representation using Eigenfaces*:

(i) You should use an array, say order, to store the indices of the eigenvalues in descending order.

(ii) Select a demeaned image Demeanface(:,i) in the training set, and project it onto the eigenvectors as follows:

```
for j = 1:N              % N is the number of eigenvectors available
   coef(i,j) = DemeanFace(:,i)'*EV(:,order(j));
end
```

(iii) Reconstruct the training image using M = 4, 8, … eigenfaces as follows:

```
for j = 1:M
   ReconstImage = ReconstImage+coef(i,j)*EV(:,order(j));
end
ReconstImage = MeanFace + ReconstImage;
```

Display the reconstructed images using different values of M, and compute the corresponding differences to the original image in terms of the mean squared error. Discuss your results.

Note that the sum of squared error between two images A and B can be computed as follows:

```
Difference = A – B;
SSE = sum(sum(Difference.*Difference));
```

(iv) In the folder “Testing”, two face images and two non-face images as shown in Fig. 2 are stored. Project these images to the eigenspace and then reconstruct them with all the eigenfaces available. Display the original images and the corresponding reconstructed images, and compute their errors. Discuss your results.

1.bmp 2.bmp 3.bmp 4.bmp

Figure 2. The four testing images.

(e) *Face Recognition using Eigenfaces*:

Suppose the coefficients of projecting the two testing face images are stored in TCoeff(i,:), where i = 1, 2. Compute the Euclidean distances of the projection coefficients between the testing images and the training images.

Display the two testing images and their corresponding three training images which have the smallest computed Euclidean distances. Discuss your results.

– END –